CoNe Freiburg

IIF INSTITUT FÜR INFORMATIK FREIBURG

Systeme II

7. Die Datensicherungsschicht (Teil 2)

Thomas Janson°, Kristof Van Laerhoven*, Christian Ortolf°

Folien: Christian Schindelhauer°

Technische Fakultät

°: Rechnernetze und Telematik, *: Eingebettete Systeme

UNI FREIBURG

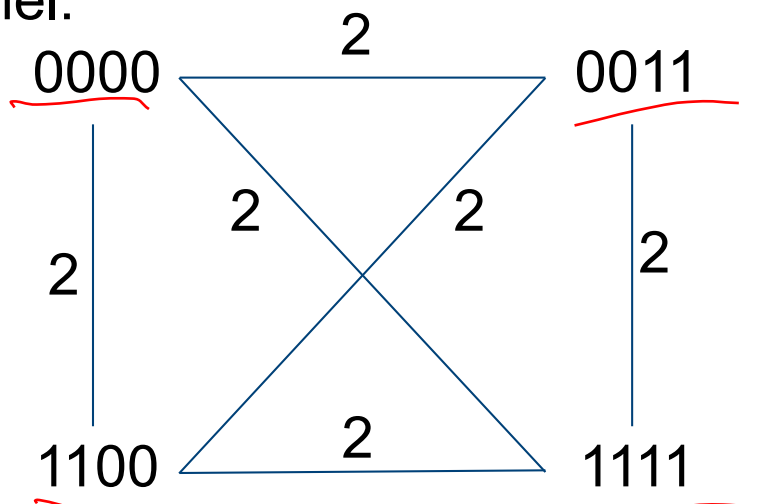
Hamming-Distanz von Nachrichtenmengen

- Die Hamming-Distanz einer Menge von (gleich langen) Bit-Strings S ist:

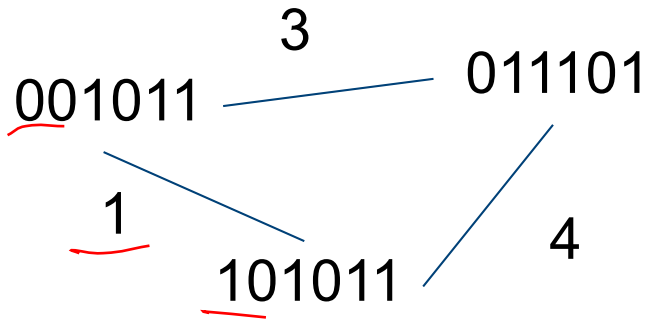
$$d(S) = \min_{x,y \in S, x \neq y} d(x, y)$$

  - d.h. der kleinste Abstand zweier verschiedener Wörter in S

Beispiel:



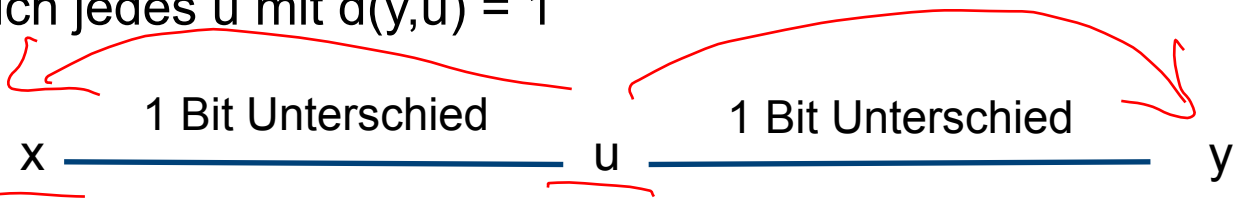
Alle Abstände sind 2



Ein Abstand ist 1!

Erkennung und Korrektur mit Hamming-Distanzen

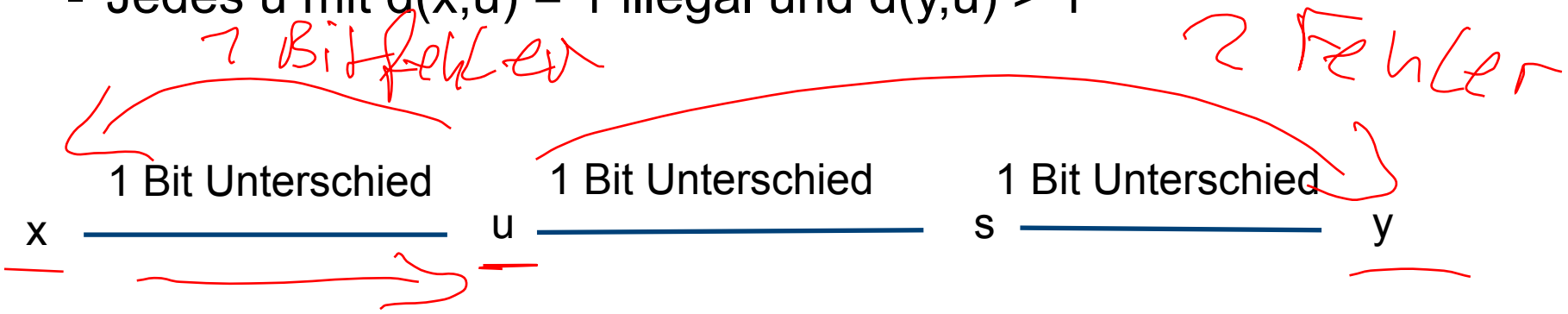
- 1. Fall d(S) = 1
  - Keine Fehlerkorrektur
  - Legale Frames unterscheiden sich in nur einem Bit
- 2. Fall d(S) = 2
  - Dann gibt es nur x, y ∈ S mit d(x,y) = 2
  - Somit ist jedes u mit d(x,u) = 1 illegal,
    - wie auch jedes u mit d(y,u) = 1



  - 1-Bit-Fehler
    - können immer erkannt werden
    - aber nicht korrigiert werden
    - z.B. Parity Bit

Erkennung und Korrektur mit Hamming-Distanzen

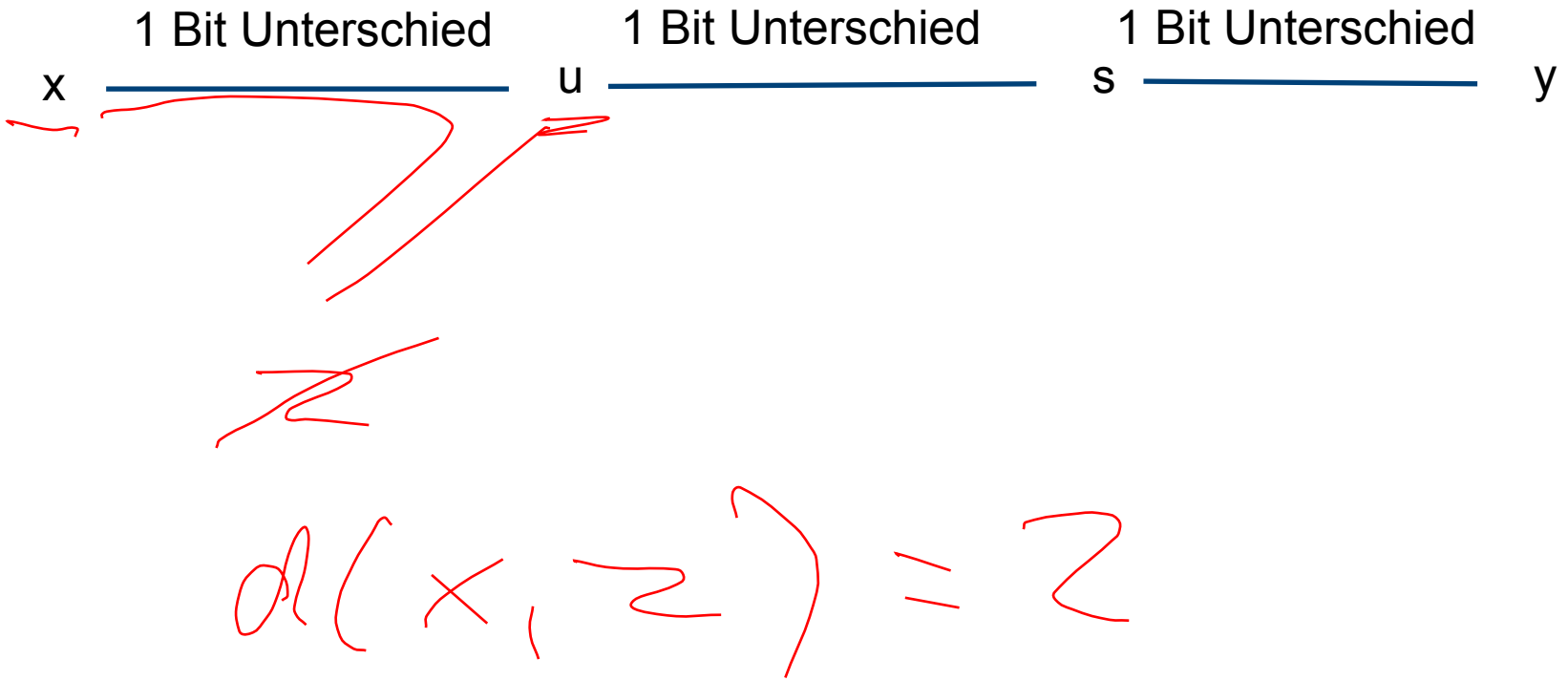
- 3. Fall d(S) = 3
  - Dann gibt es nur x, y ∈ S mit d(x,y) = 3
  - Jedes u mit d(x,u) = 1 illegal und d(y,u) > 1



  - Falls u empfangen wird, sind folgende Fälle denkbar:
    - x wurde gesendet und mit 1 Bit-Fehler empfangen
    - y wurde gesendet und mit 2 Bit-Fehlern empfangen
    - Etwas anderes wurde gesendet und mit mindestens 2 Bit-Fehlern empfangen
  - Es ist also wahrscheinlicher, dass x gesendet wurde, statt y

Erkennung und Korrektur mit Hamming-Distanzen (2)

- 3. Fall d(S) = 3
  - Dann gibt es nur x, y ∈ S mit d(x,y) = 3
  - Jedes u mit d(x,u) = 1 illegal und d(y,u) > 1



Erkennung und Korrektur mit Hamming-Distanzen

- Um d Bit-Fehler zu erkennen ist eine Hamming-Distanz von d+1 in der Menge der legalen Frames notwendig

- Um d Bit-Fehler zu korrigieren, ist eine Hamming-Distanz von 2d+1 in der Menge der legalen Frames notwendig

Codebücher und Kodierungen

- Die Menge der legalen Frames $S \in \{0,1\}^n$ wird **das Code-Buch** oder einfach Kodierung genannt.
- Die Rate R_S eines Codes S ist definiert als

$$R_S = \frac{\log |S|}{n}$$

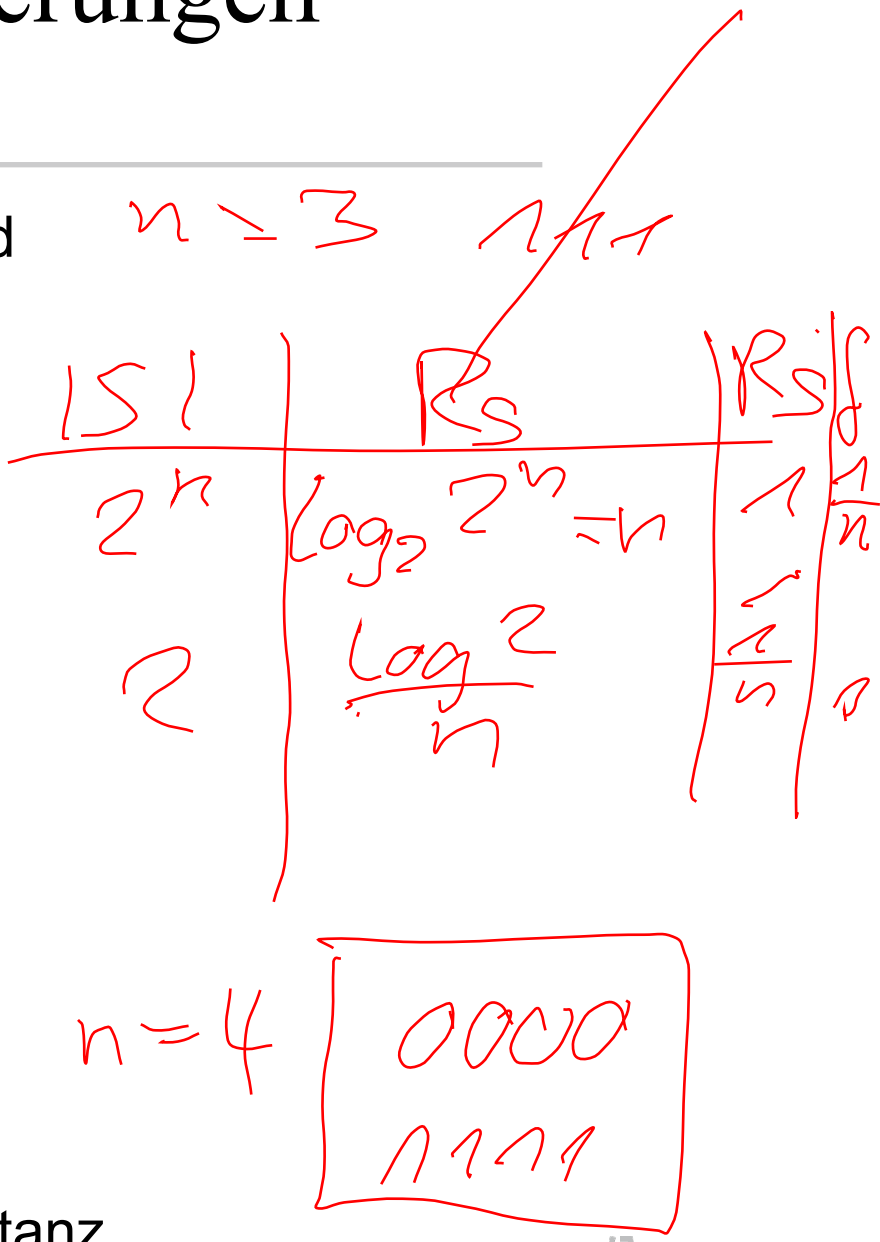
und charakterisiert die Effizienz des Codes

- Die Distanz δ_S des Codes S ist definiert als

$$\delta_S = \frac{d(S)}{n}$$

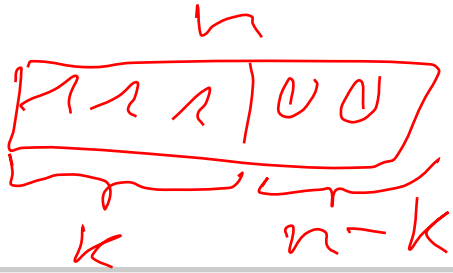
und charakterisiert die Fehlerkorrektur oder Möglichkeit der Fehlererkennung

- Gute Codes haben hohe Raten und hohe Distanz
  - Beides lässt sich nicht zugleich optimieren



CoNe Freiburg

Block-Codes

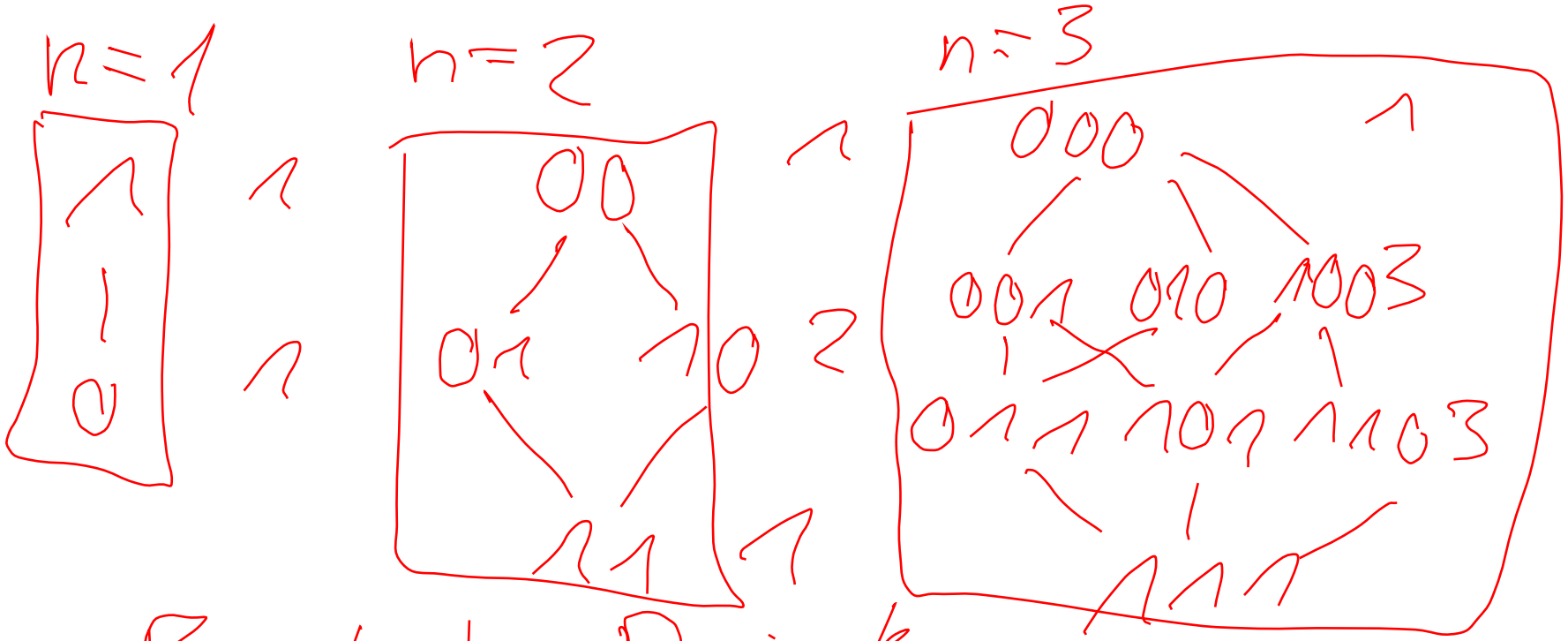


- Block-Codes kodieren *k* Bits Originaldaten in *n* kodierte Bits
  - Zusätzlich werden *n-k* Symbole hinzugefügt
  - Binäre Block-Codes können höchstens bis zu *t* Fehler in einem Code-Wort der Länge *n* mit *k* Originalbits erkennen, wobei (Gilbert-Varshamov-Schranke):

$$2^{n-k} \geq \sum_{i=0}^{t} \binom{n}{i}$$

  - Das ist eine theoretische obere Schranke
- Beispiele
  - Bose Chaudhuri Hocquenghem (BCH) Codes
    - basierend auf Polynomen über endlichen Körpern (Galois-Körpern)
  - Reed Solomon Codes
    - Spezialfall nichtbinärer BCH-Codes

Block-Codes (Herleitung 1)

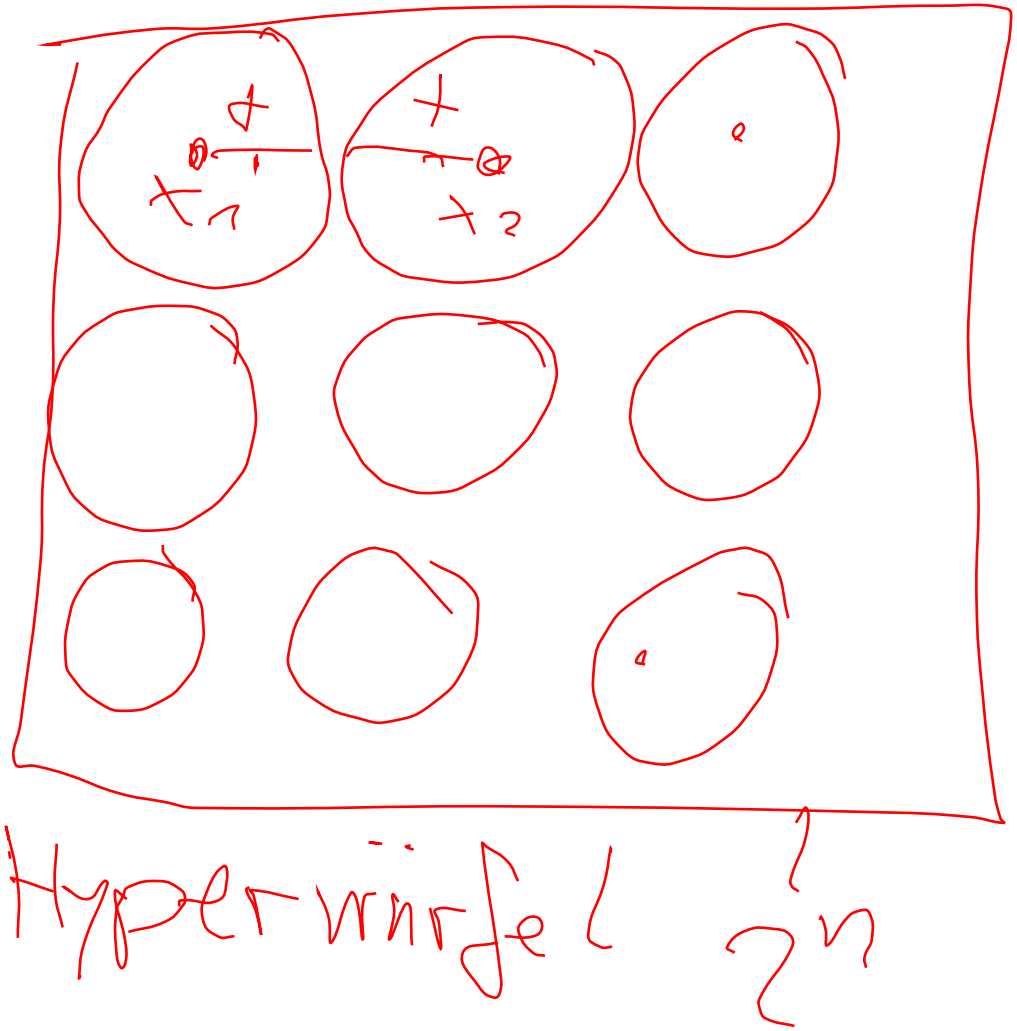


Pascalsche Dreieck

n=	1	2	3	4	← (n k)
k 0	1	1	1	1	
1	1	2	3	4	
2		1	3	6	
3			1	4	
4				1	

$$\sum_{i=0}^{n} \binom{n}{i}$$

Block-Codes (Herleitung 2)



$$\frac{2^n}{\sum_{i=0}^{+} \binom{n}{i}} = \#\text{Codes}$$

$$\sum_{i=0}^{+} \binom{n}{i} = 2^k$$

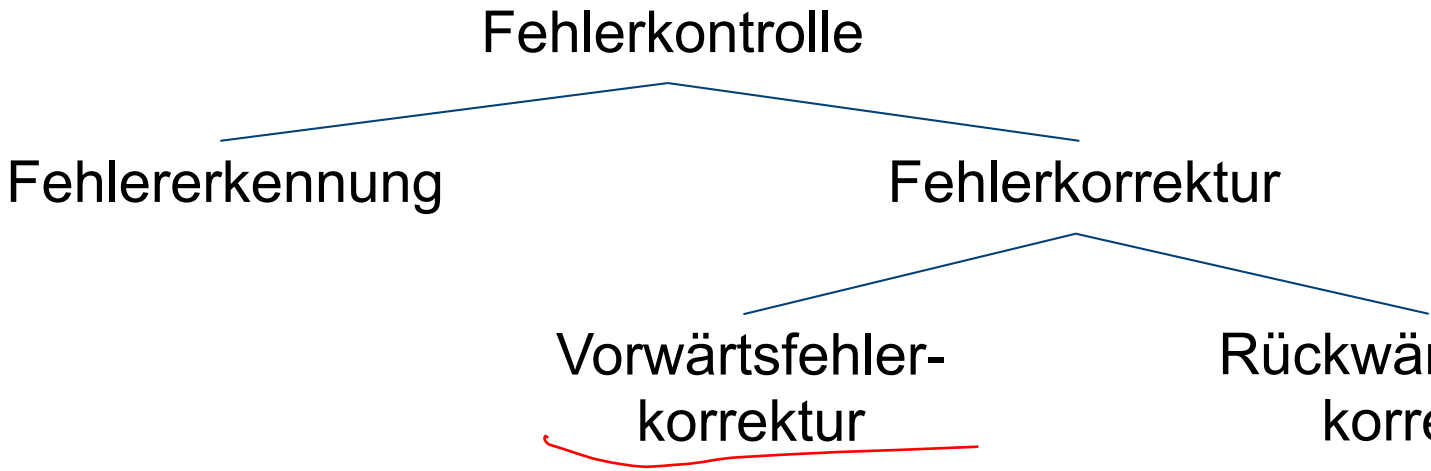
$$2^n = \sum_{i=0}^{+} \binom{n}{i}$$

$$\frac{2^n}{2^k} = 2^{n-k}$$

Fehlerkontrolle

CoNe Freiburg

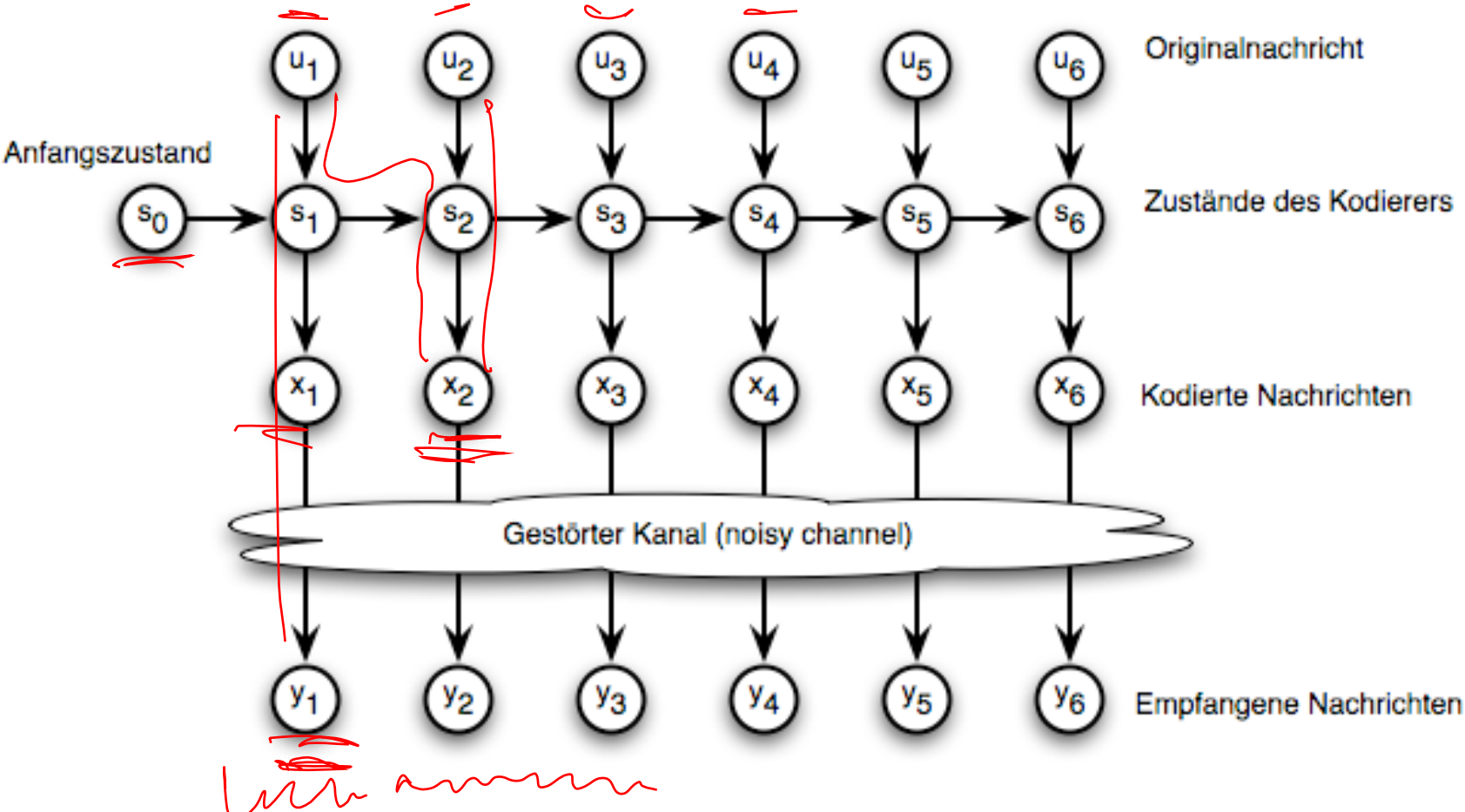
- Zumeist gefordert von der Vermittlungsschicht
  - Mit Hilfe der Frames
- Fehlererkennung
  - Gibt es fehlerhaft übertragene Bits?
- Fehlerkorrektur
  - Behebung von Bitfehlern
  - Vorwärtsfehlerkorrektur (Forward Error Correction)
    - Verwendung von redundanter Kodierung, die es ermöglicht Fehler ohne zusätzliche Übertragungen zu beheben
  - Rückwärtsfehlerkorretur (Backward Error Correction)
    - Nach Erkennen eines Fehlers, wird durch weitere Kommunikation der Fehler behoben



UNI FREIBURG

Faltungs-Codes

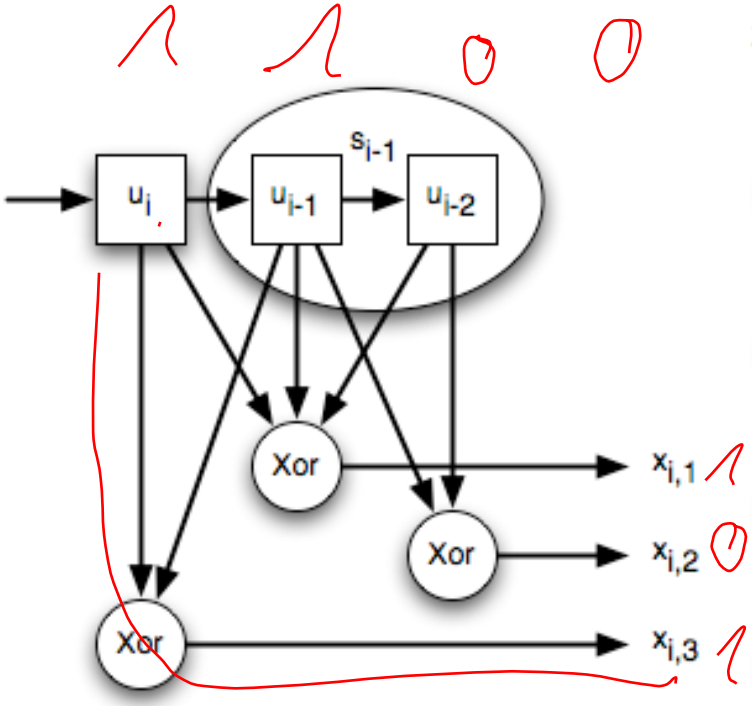
- Faltungs-Codes (Convolutional Codes)
  - Daten und Fehlerredundanz werden vermischt.
  - k Bits werden auf n Bits abgebildet
  - Die Ausgabe hängt von den k letzten Bits und dem internen Zustand ab.



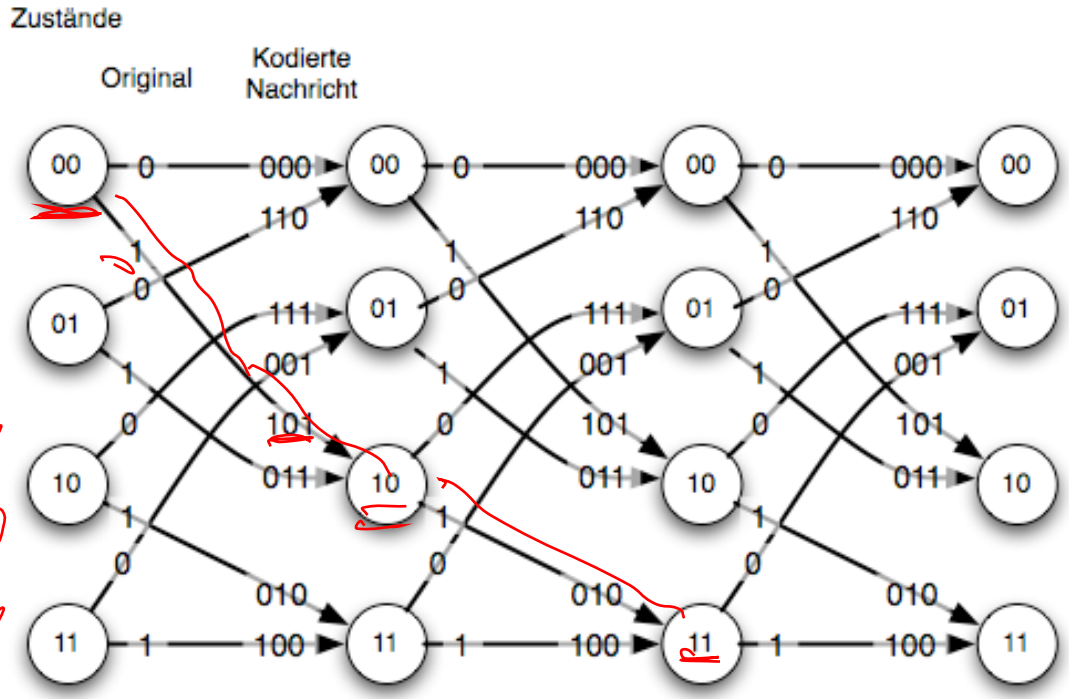
CoNe Freiburg

Beispiel

Faltungs-Kodierer



Trellis-Diagramm



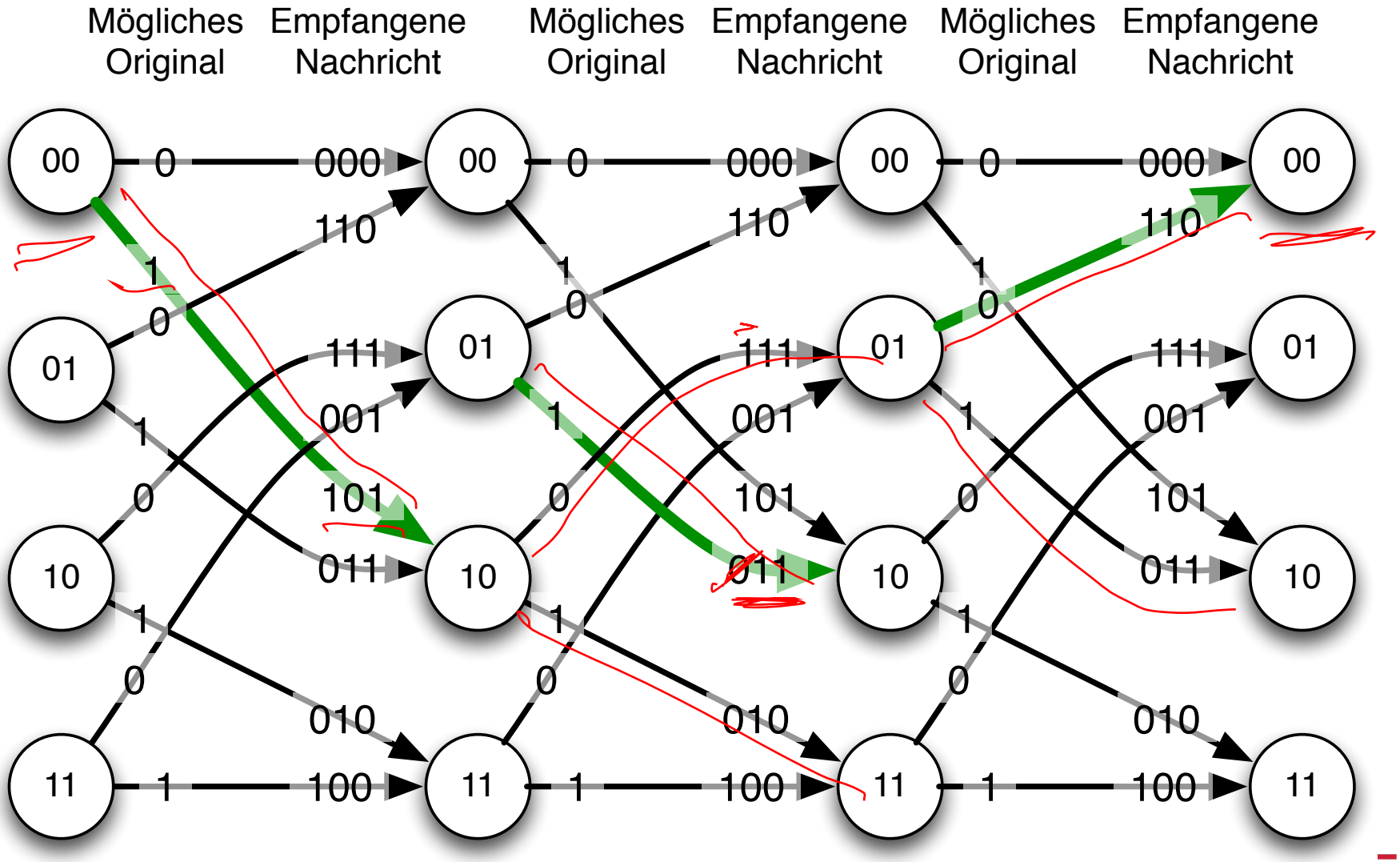
CoNe Freiburg

Dekodierung der Faltungs-Codes: Algorithmus von Viterbi

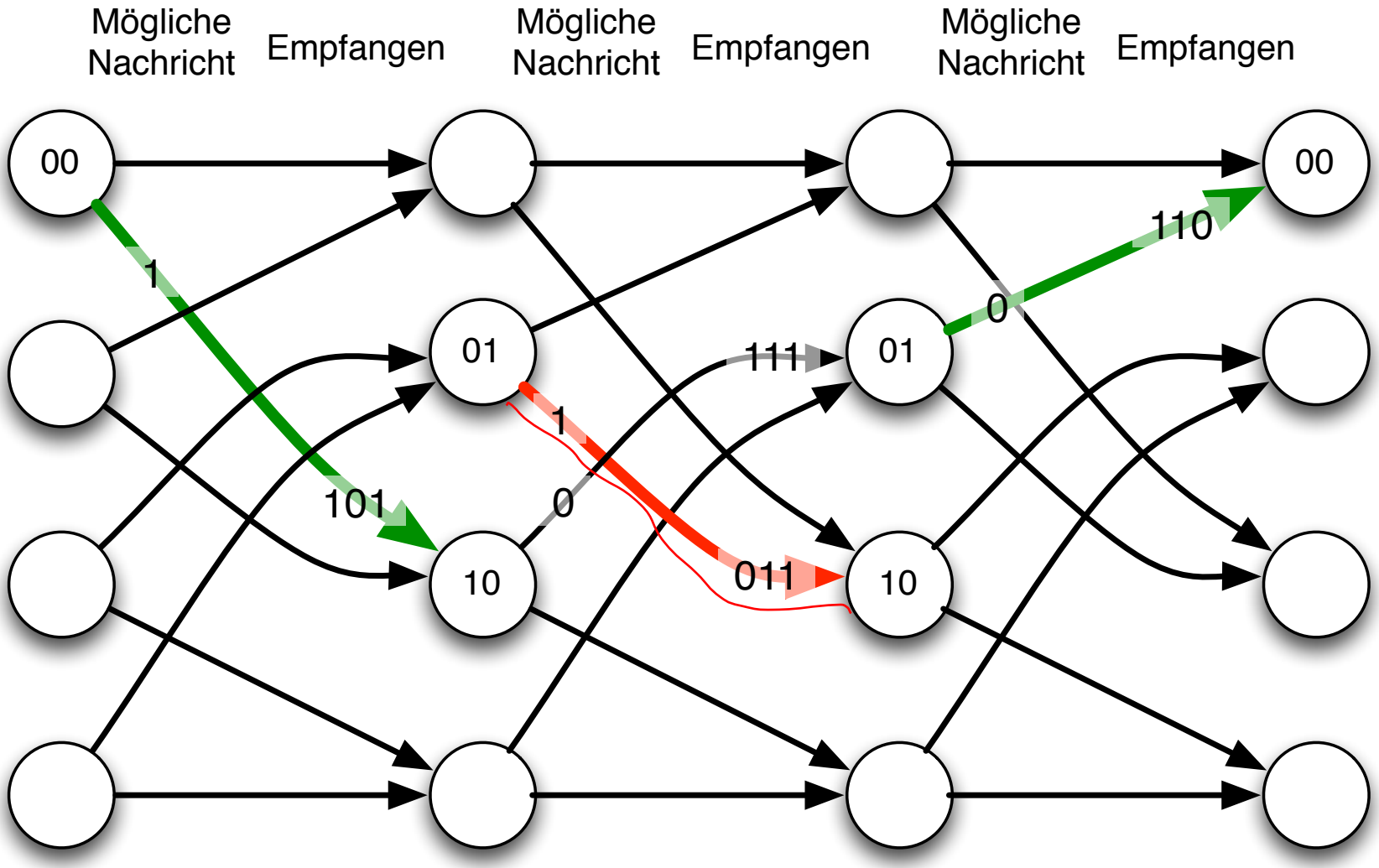
- Zwei notwendige Voraussetzungen für Dekodierung
  - (für den Empfänger) unbekannte Folge von Zuständen
  - beobachtete Folge von empfangenen Bits (möglicherweise mit Fehler)
- Der Algorithmus von Viterbi bestimmt die warscheinlichste Folge von Zuständen, welches die empfangenen Bits erklärt
  - Dynamische Programmierung
  - Hardware-Implementation möglich

Dekodierung (I)

Zustände

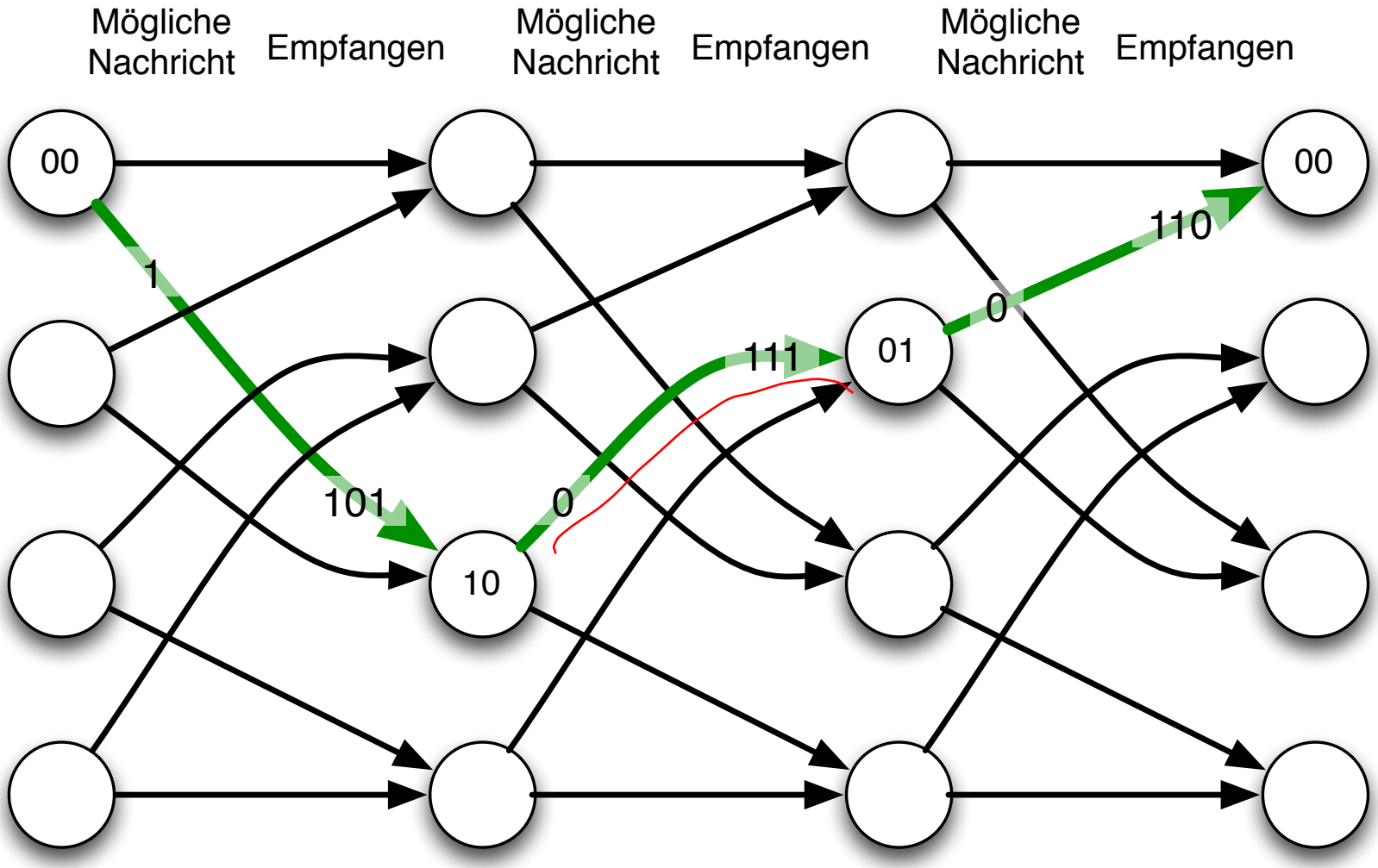


Dekodierung (II)



CoNe Freiburg

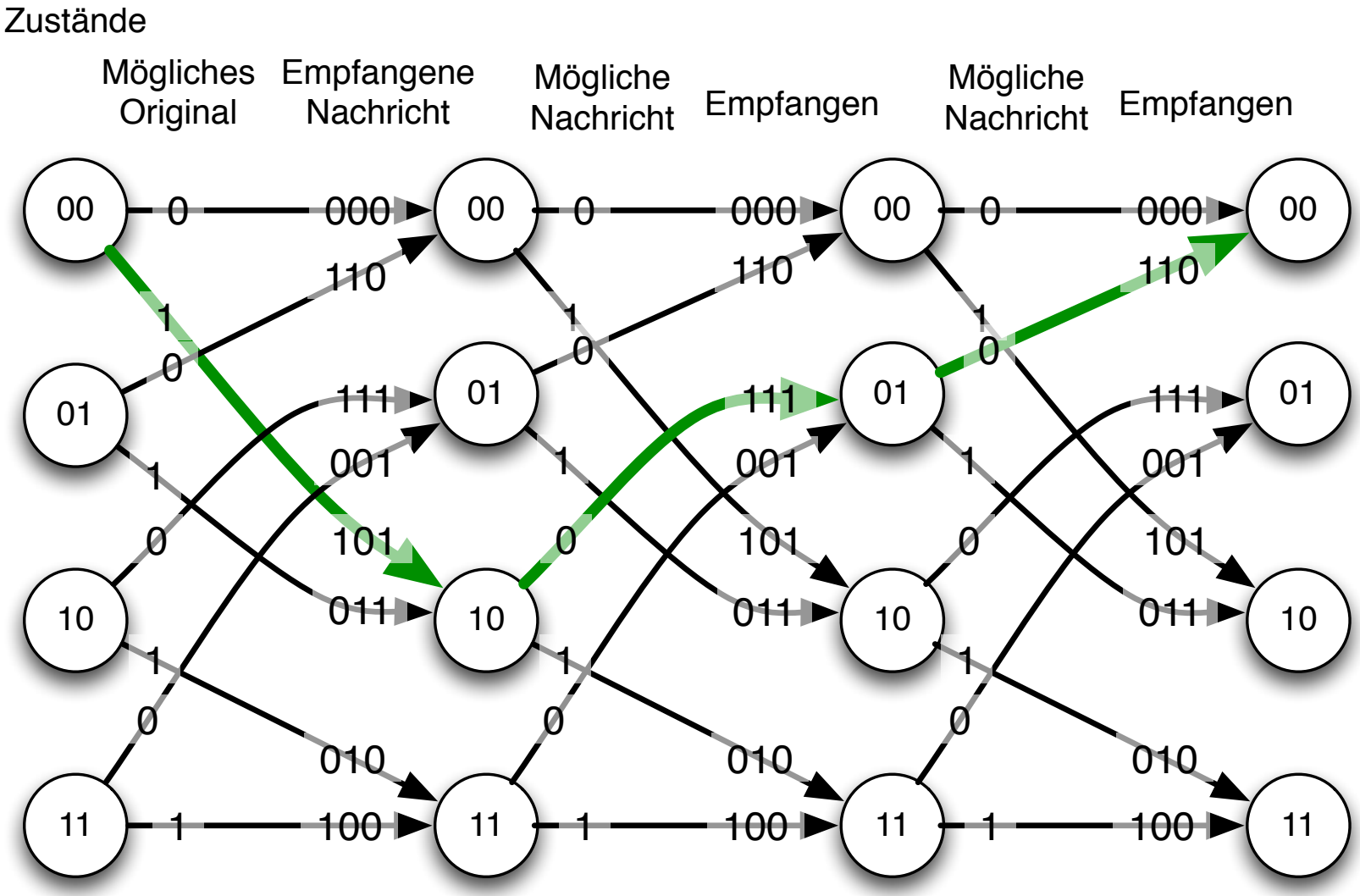
Dekodierung (III)



UNI FREIBURG

CoNe Freiburg

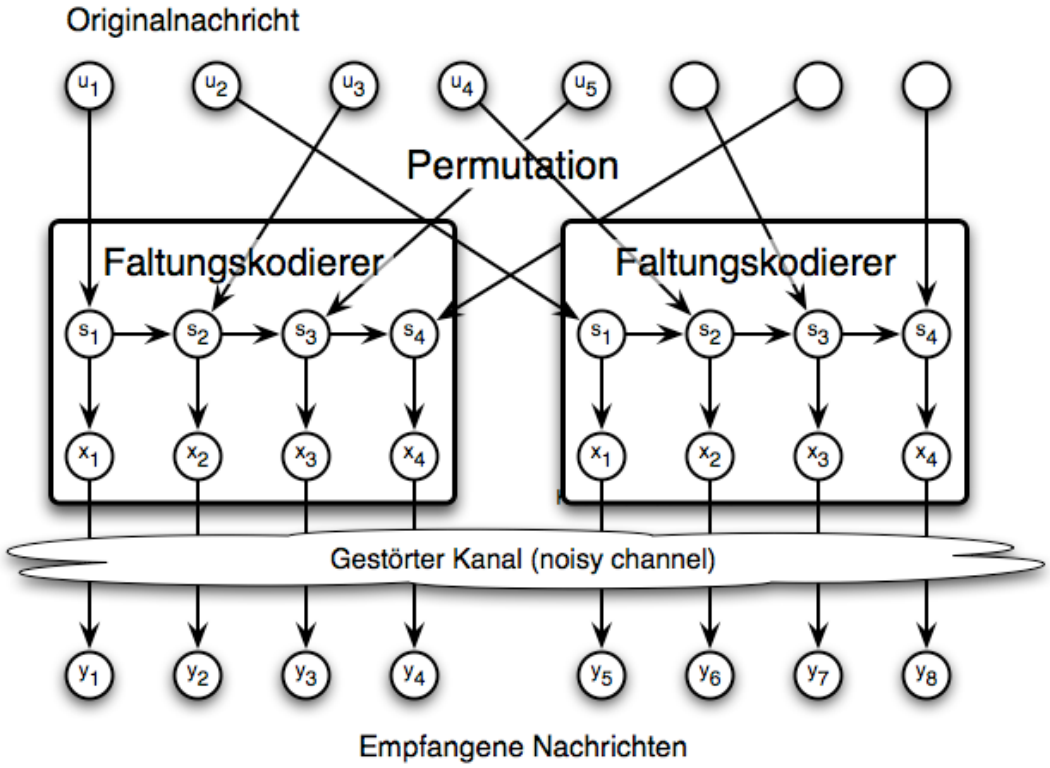
Dekodierung (IV)



CoNe Freiburg

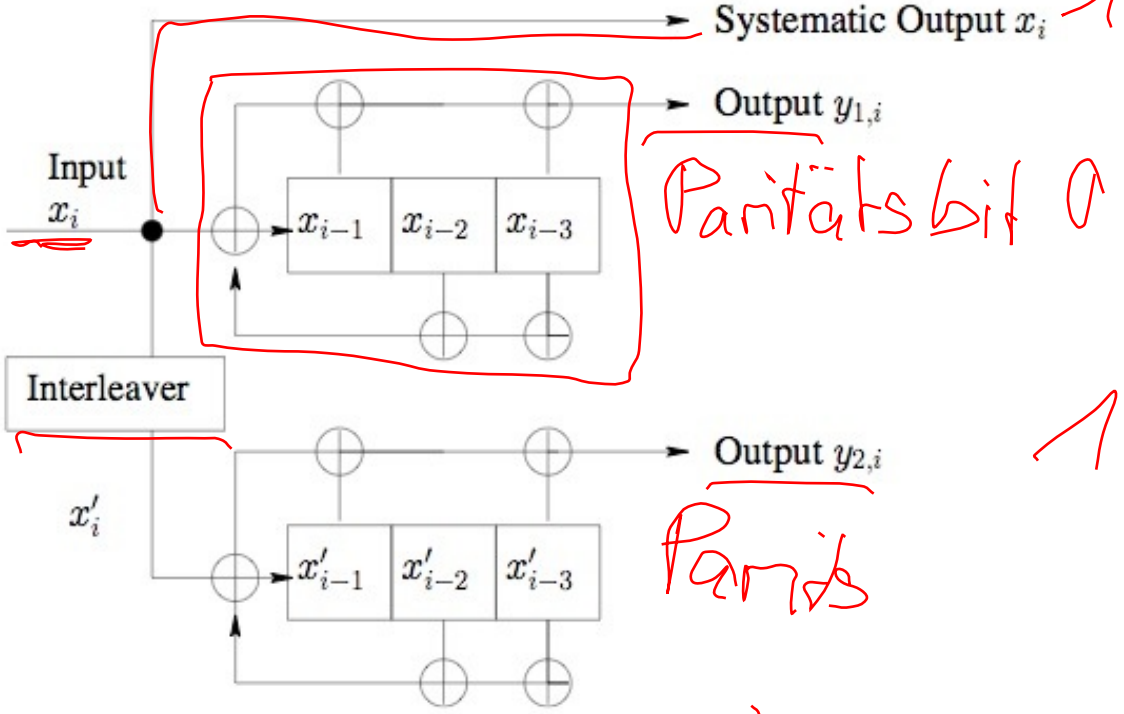
Turbo-Codes

- Turbo-Codes sind wesentlich effizienter als Faltungs-Codes
  - bestehen aus zwei Faltungs-Codes welche abwechselnd mit der Eingabe versorgt werden.
  - Die Eingabe wird durch eine Permutation (Interleaver) im zweiten Faltungs-Code umsortiert



Beispiel UMTS Turbo-Code

- Dekodierung von Turbo-Codes ist effizienter möglich als bei Faltungscodes
- Bitfehlerrate (BER) 10^{-5} bei SNR 0,7 dB
  - maximale Shannon Kapazität besser für ein 0.5 dB geringeres Signal-Rausch-Verhältnis
- Vorteile
  - Kompensation von Bursts
  - geringere Sendeleistung bzw. höhere Reichweite zugunsten geringerer Datenrate



CoNe Freiburg

Interleavers

- Fehler treten oftmals gehäuft auf (Bursts)
  - z.B.: Daten: 0 1 2 3 4 5 6 7 8 9 A B C D E F
  - mit Fehler: 0 1 2 3 **?** **?** **?** **?** **?** 9 A B C D E F
- Dann scheitern klassische Kodierer ohne Interleavers
  - Nach Fehlerkorrektur (zwei Zeichen in Folge reparierbar):
    0 1 2 3 4 5 **?** 7 8 9 A B C D E F
- Interleaver:
  - Permutation der Eingabekodierung:

    0 1 2 3
    4 5 6 7
    8 9 A B
    C D E F

  - z.B. Row-column Interleaver:
    0 4 8 C 1 5 9 D 2 6 A E 3 7 B F
  - mit Fehler: 0 4 8 C **?** **?** **?** **?** **?** 6 A E 3 7 B F
  - Rückpermutiert: 0 **?** **?** 3 4 **?** 6 7 8 **?** A B C D **?** F
  - nach FEC: 0 1 2 3 4 5 6 7 8 9 A B C D E F

UNI FREIBURG